Amendments to Specification:

Please replace the paragraph beginning on page 1, line 6, and ending at line 12, with the following amended paragraph:

The present application is related to U.S. Patent Application No. [[_____]] 09/598,534, entitled EVIDENCE-BASED SECURITY POLICY MANAGER, U.S. Patent Application No. [[_____]] 09/599,015, entitled FILTERING A PERMISSION SET USING PERMISSION REQUESTS ASSOCIATED WITH A CODE ASSEMBLY, and U.S. Patent Application No. [[_____]] 09/598,814, entitled EVALUATING INITIALLY UNTRUSTED EVIDENCE IN AN EVIDENCE-BASED SECURITY POLICY MANAGER, assigned to the Assignee of the present invention.

Please replace the paragraph beginning at page 3, line 11 and ending at line 18 with the following rewritten paragraph:

The global security method described in the exemplary approach determines whether the requested permissions have been granted to a code assembly for a given protection domain. In addition, the global security method determines whether preceding code assemblies in the runtime call stack also have the requested permission based on the protection domain of each individual code assembly. The exemplary approach does not, however, allow customized security functionality to be incorporated into the security model. Instead, the approach is static, in that the characteristics that comprise a protection domain are limited to only two parameters, which values ~~of which~~ are fixed by the security policy.

Please replace the paragraph beginning at page 8, line 5 and ending at line 16 with the following rewritten paragraph:

An embodiment of the present invention determines whether a called code frame has a ~~a~~ set of one or more permission (i.e., a requested permission) available to it, so as to be able to execute a protected operation. Typically, a code frame is contained within a code assembly received from a remote resource location (e.g., across the Internet) or local resource location (e.g., a system's hard drive, a network peripheral, or a system-attached peripheral).

Please replace the paragraph beginning on page 20, line 23 and ending at line 32 with the following rewritten paragraph:

Generally, an "intersection" operation (represented by the symbol "∩") is a set operation that yields the common elements of the operand sets. For example, if Set1 includes elements A, B, and C, and Set2 includes elements B, C, and D, then the intersection of Set1 and Set2 (i.e., Set1 ∩ Set2) equals B and C. Accordingly, the intersect() method returns the minimum set of permissions for which a demand that passes both permissions objects will also pass their intersection. The intersect() method provides a means to retrieve shared state between two permissions. If there is no shared state (i.e., permission) between the two instances, then the method returns "null". Otherwise, the method returns a new permission object with the permission-defined intersection of the original permission object and the target permission object.

Please replace the paragraph beginning on page 31, line 14 and ending at line 21 with the following rewritten paragraph:

In the illustrated embodiment, if the code frame in code assembly 502 (also referred to as the code frame 502) requests permission P4 to perform a protected operation, the permission request object 508 will walk the stack, starting with the permission grant objects associated with the code assembly 504, which contains the calling code frame (also referred to as code frame 504), to determine if the requested permission is available up the call chain. As shown by the exemplary permissions illustrated in FIG. 5, both the permission grant object 512 and the permission grant object 514 include the permission P4. Therefore, the called code frame 502 may be executed to perform the protected operation.

Please replace the paragraph beginning on page 31, line 22 and ending on page 32 at line 5 with the following rewritten paragraph:

In the illustrated embodiment, if the code frame 502 requests permission P3 to perform a protected operation, the permission request object 508 will walk the stack, starting with the permission grant objects associated with the code assembly 504, to determine if the requested permission is available up the call chain. As shown by the exemplary permissions illustrated in FIG. 5, both the permission grant object 512 and the permission grant object 514 include the

permission [[~~P4,~~]] P3, although the permission grant object 516 does not. Nevertheless, the called code frame 502 may be executed to perform the protected operation.